2019

Two Player Zero Sum Multi-Stage Game Analysis Using Coevolutionary Algorithm

Sumedh Sopan Nagrale
Wright State University

TWO PLAYER ZERO SUM MULTI-STAGE GAME ANALYSIS
USING COEVOLUTIONARY ALGORITHM.

A Thesis submitted in partial fulfilment of the
requirements for the degree of
Master of Science in Electrical Engineering

By

SUMEDH SOPAN NAGRALE
B.E.,University of Mumbai, 2012

2019
Wright State University

WRIGHT STATE UNIVERSITY

GRADUATE SCHOOL

April 24, 2019

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY Sumedh Sopan Nagrale ENTITLED Two Player Zero Sum Multi-Stage Game Analysis Using Coevolutionary Algorithm BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF Master of Science in Electrical Engineering.

Luther Palmer, III, Ph.D.

Thesis Director.

Fred D. Garber, Ph.D.

Chair, Department of Electrical Engineering

Committee on Final Examination:

Luther Palmer, III, Ph.D.

Pradeep Misra, Ph.D.

Xiaodong Zhang, Ph.D.

Barry Milligan, Ph.D.

Interim Dean of the Graduate School

ABSTRACT

Nagrale, Sumedh Sopan. M.S.E.E, Department of Electrical Engineering, Wright State University, 2019. Two Player Zero Sum Multi-Stage Game Analysis Using Coevolutionary Algorithm.

A New Two player zero sum multistage simultaneous Game has been developed from a real-life situation of dispute between two individual. Research identifies a multistage game as a multi-objective optimization problem and solves it using Coevolutionary algorithm which converges to a solution from pareto optimal solution. A comparison is done between individual stage behaviour and multistage behaviour. Further, simulations over a range for Crossover rate, Mutation rate and Number of interaction is done to narrow down the range for a range with optimal computation speed. A relationship has been observed which identifies a relationship between population size, number of interactions, crossover rate, mutation rate and computational time. A point from the obtained range is then selected and applied to a new game to see if the point from the narrowed range works.

Contents

List of Figures

List of Tables

ACKNOWLEDGMENT

If bunch of thoughts were to be considered as the initial population on the research, The guidance provided by *Dr Palmer* acted as a fitness evaluation of those thought, without his guidance this work would not have been possible.

I would like to recognize the presence and support by my family and all my friends here and back in India.

At Last, I would like to acknowledge people whose life work in science and technology has influence on my thoughts, Dr A P J Kalam, Nikola Tesla and all the scientists, researchers and explorers whose efforts made this research possible in this space-time continuum.

DEDICATION

The Thesis is dedicated to *Dr B. R. Ambedkar*, To whom I owe everything.

Chapter 1

Background

Background primarily introduces reader with the terms, topics and concepts involved in the Thesis and contemporary work in Game Theory.

1.1 Game Theory

Game Theory translates real life situations of decision making in the form of game, which can be sometime seen as a simplified representation of the same idea considering all significant components involved.[14]

Game theory has been extensively studied field mainly in economics, However if one investigates, one can find its interconnections with and application over a broad range of fields. Game theory analyses decision making of two or more players who interact with each other based on the pay-off received by their action in the game [5].

Such a strategic interactions between individuals or groups are quite common in real life,So one can see that there has been presence of Game Theory for a long time. However, The foremost formal representation of games and solutions were presented by Von Neumann and Morgenstern (1944) in their book ”Theory of games and economic behaviour” [27], which analyses zero sum game. Followed by John Nash’s dissertation [15], which establishes the presence of a Nash equilibria for a game where no player is benefited by deviated from the equilibria strategy.

In the year 1954, O.G. Haywood used game theory to analyse the military decisions used in the Battle of the Bismarck Sea, a battle fought during the World War II [10]. The predicted outcome of the game was same as the actual outcome of the war. Apart from wars Game Theory has it’s presence everywhere where one can find any sort of decision making. From Economics (In order to understand the behaviour of companies in a given market) to international relationship (Current trade war is rational on basis of Game theory) to Engineering (Coordination of Robots) to Medical Science to a problem such Co_2 emission. Another example will be bargaining at a shop. Each player involved in the game is rational enough to make his profit maximum. (seller trying to sell it at maximum price, buyer trying to buy at minimum price.).

In order to understand more about games and an overview will be followed of the common and essential terms which are used in Game Theory. This serves two fold purpose, firstly,it gives an clear understanding of the terms used in the research for the individuals new to the topic and secondly, it removes ambiguity if any that would have been presented in the text otherwise.

Terms

1. Action: Actions are the choices available for a player to select from while playing a game.
2. Move: Actual Action selected from the action set is called Move.
3. Play: The entire sequence of Moves for a player from initial point to the final state of the game is called play.
4. Terminal State: Terminal State is the state from where a game does not move forward.
5. Strategies: Strategies are the set of action or just an action that a player takes while playing a game. Strategies may be deterministic, probabilistic, conditional or behavioural.
   (a) Pure Strategy: Pure strategy is a deterministic strategy when a player plays same action set each time it plays the game.
   (b) Mixed Strategy: Mixed strategy is a probabilistic strategy which assigns a probability distribution to the pure strategy. This means the player will make a move with certain probability. This means player may make different move when the game is played.
   (c) Conditional Strategy: Conditional strategy is the one in which player decide their strategy in next stage based on the strategy of the other player on the current stage.
   (d) Behavioural strategy: Mixed strategy is the one in which probability distribution is applied to the pure strategy.Which is also called as Global randomization.Whereas, Behavioural strategy assigns probability to individual information set (for more information refer extensive form representation describes in the next section), such that the actions available will follow that probability distribution.
   (e) Dominates:(Dominant)[7] An action B dominates A : If B gives as good as or better payoff than A for opponent's actions.(B$\geq$A)
      i. B strictly dominates A: Action B always gives better payoff than A.($B > A$)
      ii. B weakly dominates A: There is at least one set of opponents' action for which B is superior, and all other sets of opponents' actions give B the same payoff as A.($B > A$ for at least one,B=A, for rest).

[Strategy B is strictly dominant: If strategy B strictly dominates every other possible strategy. Strategy B is weakly dominant: If strategy B dominates all other strategies, but some (or all) strategies are only weakly dominated by B.]

6. Rationality: A player playing a game can be rational or irrational. A player is rational if the player plays for profiting itself, Whereas an irrational player could simply play with random guess with particular aim of maximising its profit.

   (a) Individual Rationality:
   A player is said to be rational if it maximises its expected pay-off.

   (b) Group Rationality:
   In case of coalition games, where players form coalition. A group of players are called as group rational or efficient when the total payoff (does not exceeds total eligible payoff available) is equal to the sum of the payoff received by the individual players

7. Information: knowledge a players have about the game i.e about the available actions, number of players and about the decision it took.
   Complete Information: All players know everything about each other. The following is the common knowledge

   (a) Players who are playing the game

   (b) Their utilities.

   (c) Players strategies

   (d) Type of players

   Incomplete information: Information players have are

   (a) His true payoff

   (b) Other player's mean payoff (beliefs or knowledge of payoff with small random fluctuations) and player's Identity.

   Information player may not know

   (a) Type of other player

   (b) Strategies of the other player

   (c) Exact payoff of other player.(knows mean payoff)

   (d) rules of the game are not well defined (strategy unknown)

   Higher Order of incomplete information:
   Rules of the game can be considered as higher order of incomplete information. Different players may have beliefs about the game that is being played.

Game Classification

1. Based on Moves

   Depending upon the moves of the game we have

   (a) Simultaneous Game:

       All players make their move simultaneously. These are games of imperfect information. Player doesn't know what action is taken by other player.

   (b) Sequential Game:

       Players make one move after other in a sequence. Sequential games in which player can not observe the action taken by other player can also be treated as simultaneous game.

2. Based on Payoff

   Depending on the payoff each player receives for playing the game.

   (a) Zero sum Game: This are games of pure conflict. This means one's loss is other player's profit. The sum of payoff received by the players is equal to zero.

   (b) Constant sum Game: The sum of payoff received by the players is equal to constant.

3. Based on Information

   (a) Perfect Information: Player knows previous decisions of all players including his own before he has to make the next move. In such a case the game is of perfect information.

   (b) Imperfect Information: Player forgets what decision he took earlier in the game or he does not know decision that is being made by other player (as in case of simultaneous game). However, they know: who the other player are, their strategies and payoffs. Information about other player in imperfect information may be complete.

Game Representation

Game can be represented in various ways, depending upon the information we have regarding the game. Game is defined based on Number of players, Actions available for those players, Payoff for the players and Rules applicable to the games. Using this definition a game can be represented in various forms, representation are based on the information available to players. A Game can be represented using the following forms:

1. Normal Form

2. Extensive Form

3. Beyond Normal/Extensive Form

   (a) Stochastic Games

   (b) Bayesian Games

   (c) Congestion Games

   (d) Multistage Games

(e) Repeated Games

Normal form Games: It describes all possible strategies and utility of the agents against eachother. It shows strategic interactions between the players. It consists of

1. Set of players
2. All strategies of the players
3. Payoff received by players for the actions.

An Example of Normal Form representation for Rock-Paper-Scissor game

Extensive Form Games

		Player 2		
		R	P	S
Player 1	R	$0, 0$	$-1, 1$	$1, -1$
	P	$1, -1$	$0, 0$	$-1, 1$
	S	$-1, 1$	$1, -1$	$0, 0$

Table 1.1: 3x3 Matrix: Rock-Paper-Scissor Normal-Form Game

Extensive Form is a tree like structure. It consists of

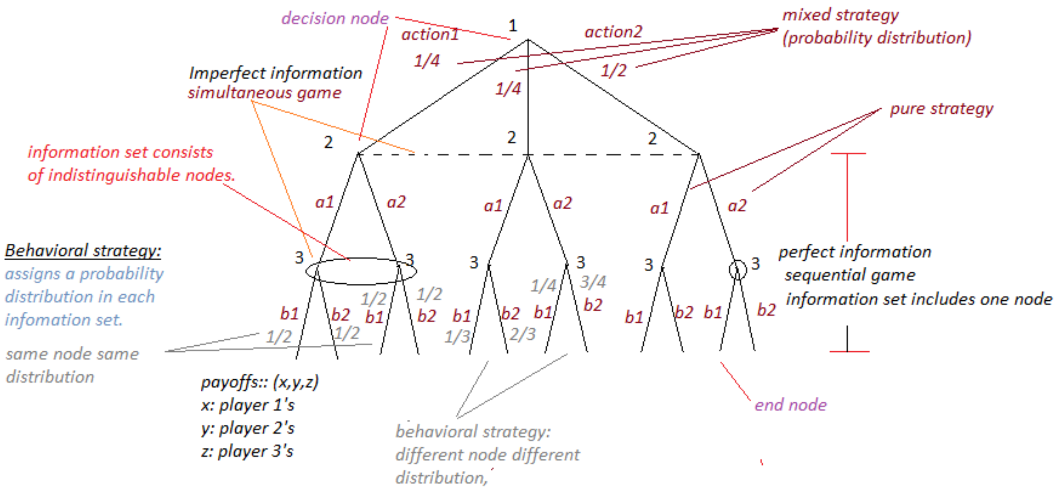
1. All Players
2. Player's Strategy mixed or pure
3. Payoff received by players for the actions

It gives detailed information about the game such as

1. Simultaneous or sequential game
2. Complete or incomplete information.
3. Perfect or imperfect information
4. Pure or mixed strategy.

Figure 1.1 describes the details of the extensive form games where few of the common terms used have been explained.

1. Decision Node: Where a player chooses an action.
2. Chance Node: Where Nature or chance chooses an action
3. End Node: Where there are no more decision to be made or simply the last node.
4. Information set: Information set consists of nodes for a player, Information set may include one or more nodes. Players are unable to distinguish between the nodes in the same information set.



Extensive form representation

Beyond Normal-Extensive Form

1. Multistage Game A multi-stage game is a finite sequence of stage-games, each one being a game of complete but imperfect information (a simultaneous move game).
   For a Multistage game
   (a) Each stage game is complete and with imperfect information.
   (b) All players observe the outcome of each stage game, This is common knowledge for the players
2. Repeated Games: A special case of multistage game is the one in which same game is played over and over again in each stage. A repeated can be of one of the following types
   (a) Finitely Repeated Games
   (b) Infinitely Repeated Games
3. Stochastic Games: Stochastic games are also called as Markov games. There are two differences between MDP and Stochastic Game is the number of decision maker.
   (a) In case of MDP we had single decision maker, however in case of Stochastic games we have multiple decision maker.
   (b) We change state in MDP, However in Stochastic games we change Stages (games) So Markov game is also a multistage game.
4. Bayesian Games: Bayesian games are the one with the incomplete information. Players select their strategies according to Bayes's rule.
5. Congestion Games: Players playing the game will have same strategies available.

1.2 Solution Concept

Rationality of the player plays an important role in the development of a solution which could be termed as optimal or beneficial. The solution of a game is the set of strategies (action taken by the each player in the game as in totality) depends upon multiple factors which can be stated below

1. Are the Players Rational or Irrational?
2. Up to what extent players have knowledge about game?
3. Is the game a Complete or incomplete game?
4. Is the game Perfect or imperfect game?

1.2.1 Minimax Theorem:

Minimax theorem of von Neumann and Morgenstern says that, given a two-person zero-sum game, there is always a pair of strategies, either in pure or mixed strategies, such that the maximin payoff equals the minimax payoff of player 1.In two person zero-sum games the maximin payoff of player 2 with respect to his own payoff values is identical to the minimax value with respect to the payoffs of player.

1.2.2 Nash Equilibrium

For a finite game, with finite number of players with finite strategies each, there exists an equilibrium strategies such that no one player can get higher payoff by deviating from the equilibrium strategies.This strategic equilibrium consist of best response of players and is called as Nash equilibrium. Nash equilibrium hinges on two assumption of a player

1. Other players are rational
2. Other players chooses the equilibrium such that it is the best response.

1.2.3 Rationalization

Rationalisation is the process which involves iterative deletions of such strategies which won't be the best response to any belief of the other player's move.[19] [7] It hinges on:

1. Other players are rational.
2. Rationality is common knowledge.

A strictly dominated action is a never-best response and state that a rational player won't play it if he is rational. A set of rationalize action is obtained by Iterated Elimination of these strictly dominated actions.If all the players are left with one strategy at the end of iterated elimination of dominated action then the game is called a dominance-solvable game.
Note:

1. Nash equilibrium is a rationalizable equilibrium; however, the inverse is not true.Thus Rationalizability is a Generalization of Nash Equilibrium.
2. An action that is not rationalizable can never be a best response to any opponent's strategy (pure or mixed)
3. The variation of the process includes deletion of both strict and weakly dominated strategies. The sequence in which the weakly dominated actions are deleted will reflect on the rationalized actions.

Zero sum Game

1. In case of zero sum game, Nash equilibrium is same as Minimax theorem.
2. In two person zero-sum games the maximin payoff of player 2 with respect to his own payoff values is identical to the minimax value with respect to the payoffs of player.
3. If the payoff values of the two players in each cell add to the same constant value, then the game is equivalent to a zero-sum game and can, without loss of information, be transformed into such a game.
4. There is atleast one mixed Nash equilibrium in a game.

Multistage Game:

As stated earlier repeated game is a special case of multistage game. A multistage game is the one in which all the stages may not be different. Each stage is complete with imperfect information. All players can observe the outcome of the game which is the common knowledge of the game.[26]

Solution to a multistage game

Consider a Multistage Game with N-stage games where each stage has Nash equilibrium as $N_1, N_2...N_N$. The sequence of outcomes determines total payoff received by an individual. which is also called as strategic link. Players may use this strategic link to gain more payoff. This may led to a solution which is not always a Nash equilibrium of the stage game.

Discounting or impatience factor: Discounting: The discounting factor defines importance given by the players on the future rewards comparable to present terms. Usually it is a convention that payoffs which are further away in the future are worth less than the payoffs obtained earlier in the sequence of play, However, this is not necessarily be true. This is the notion of discounting, or impatience, means that todays game is played now, and is certain to occur, but there is some uncertainty about tomorrows game actually occurring. Say, If two players are playing a game in period one, there may be some probability δ ¡ 1 that tomorrows game will indeed be played, however, with probability 1 δ it will not. This causes the following scenario

1. Utility = u ,if game is played with probability δ.
2. Utility = 0 ,if game is not played.

$$Expected_{utility} = \delta u$$

For a N stage Multistage game in 1, 2, ..., T. for δ = 1, future is as important as present whereas if δ ¡ 1, present is more valued than future game.

1. What is the need to deviate from Nash equilibrium?

   Nash equilibrium gives a solution which is a best reply to each player.However, This does not mean to be the payoff with highest value. So players may wish to deviate from the Nash equilibrium and obtain a higher payoff.

2. What influences this selection of Non-Nash solution strategies?

   Two elements that are crucial to support certain strategies in the stages of game(early periods in general) which is not a Nash equilibrium. Multiplicity of Nash equilibrium in 2nd stage:

   There must be at least two distinct equilibria in the second stage: which can act as a

   A) Stick (punishment::low payoff): Stick ensures no one deviates from the proposed path of play for short term gains. Sticks leads to a long term loss, if deviated.

   B) Carrot (reward::higher payoff): Carrot motivates to stay on the proposed path of play.

   The discount factor has to be large enough:

   The long term losses must be large enough to discourage any deviation, As we know this is controlled by discount factor. So a discount factor must be large enough to avoid any deviation.

1.3 Contemporary

As described in the work of Holler [11], in which he views game theory into three different stages depending on the rationale of the stakeholders involved in the game. These three stages are Classical, Modern and New Game theory. In case of classical stage, individual rationality is considered, In Modern, player believes not only he is rational but the other player involved in the process are rational to an extent where Nash equilibrium is possible and New in which player knows little about the other player, i.e other player's rationality, it forms his beliefs of the other player. Repeated games have been analysed in [2] [16] . A repeated game is a special case of Multistage game where each stage game is same. Traditional solutions involves presence of carrot and punishment in the payoff to deviate from Nash and gain better payoff which is described above. Apart from this, techniques described in [6] [13] [16] [28] [23] [1] can be used to solve the game and further , an existence of Nash equilibrium for infinite repeated games has been discussed in [21].Genetic algorithm has been used to find a solution for the game theoretic problems. The research is described in [4], [3], [9]

2

Introduction

The purpose of the research is to model real life competitive situations such as fight between individuals or companies or countries in the form of game and solve it in order to predict the outcome based on the rational optimal strategy.

The game developed on such a competitive situation is a zero sum simultaneous multi-stage game. This is a multi-stage game since a move causes a transition from one stage to another. Most of the contemporary work is done on single stage game or a repeated game as described is a special case of Multi-stage game where all stages are same.

This research mainly focuses on Multistage game with different stages where individual should have capabilities to play under all the stages. It is seen from the work of "Game Theory Based Coevolutionary Algorithm",[24] a CoEvolutionary algorithm can be applied to mimic the process of natural selection within the individuals and between the individuals in order to find a solution to the game for a single stage game. Here developed game is a multistage game which results in a Multi-objective optimisation problem instead of single objective optimisation problem. This problem is overcome by simply treating composite function as a single function.

First, Individual stages have been analysed treating them as a single-objective optimisation problem. The behaviour of the agents are observed and then compared with the behaviour corresponding to multi-stage setting. The game theoretic solutions shows that the zero sum game has infinite solutions set spread over certain actions, which reflects on simulations of Individual stage case where convergence occurs over a range of strategies. However this is different in case of multi-stage game where algorithm converges to a single strategy.

Further, A novel mathematical relationship has been developed based for coevolutionary which provides a relationship between population size, Interactions, crossover rate, mutation rate and computational time. This relationships holds good irrespective of game under consideration since it is based on algorithm rather problem specific, however depending on the selection techniques used in coevolution, equation may be needed to be modified to suit the situation better. This equation can predict time for the game without simulations.

A dataset is obtained based on simulation over a range of Mutation rate, Crossover rate and interactions for the multi-stage game. This along with the equation is then used to identify a set of parameters which can be used for games developed on the similar lines of the current game. This set of Parameters is then applied to a new game and hypothesis of reuse of parameters has been verified however with a cautious note that stages must be of similar nature i.e must have a common solution set of optimal strategies.

At Last, A brief commentary on the game is done with respect to the Multiple task learning, and how the game reflects the learning behaviour of an intelligent agent along with the discussion on future research work.

The remaining of the chapter adds on to the research described above, a brief overview of the chapter and their contents is as follows

1. Game 1: Attacker's Game or Horizontal Movement Game:
   Game description, its representation and game theoretic solution.
2. Methodology:
   Includes reason to use Coevolutionary algorithm along with the details such as selection technique and fitness calculation
3. Simulation:
   The application of coevolutionary algorithm to the game for individual cases and multiple stages and their comparison. Change of parameters and its effect on the convergence, Mathematical derivation for the relation between computational time and parameters
4. Game 2: Future Extension Movement Game:
   Game 2 description and simulation using a point from selected parameter range.

3

Game 1: Attacker's Game or Horizontal Movement Game

A real life situation is simplified into the form of a game. The basic concept of the game is the dispute between individuals or group. Horizontal Movement Game or simply Attack Game consists of two individuals which can have one of the following movements from the action set $\{Right, Left, Block, Attack\}$. As the name suggests the game is restricted to the horizontal movement.Game is a deterministic game, i.e is the one in which if one takes certain actions he will reach to a unique result. There is no randomness in the process.

3.0.1 Players

Player $\triangle$and Player $\triangledown$.

3.0.2 Environment

Zero sum Two player deterministic game is played in the following environment. There are three positions in the grid possible for the player to be in

Game I

3.0.3 Actions

Actions available for the players are

1. Right : Player moves to the right
2. Left : Player moves to the left

3. Block : Player tries to block the anticipated attack

4. Attack: Player Attacks.

The player is restricted to move in upward or downward direction.

3.0.4 States

Based on the players($\bigtriangledown$ and $\bigtriangleup$) and *Action set* (R,L,B,A) the number of states are defined as follows: The game consists of the following environment

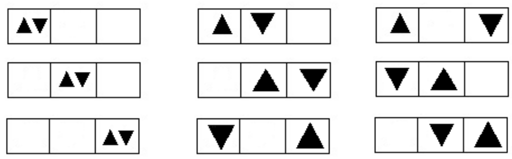


Figure 3.1: Game I States and their equivalent representation s_0,s_1,s_2

3.0.5 Representation choice

Conventionally, the states would have been defined in the terms of relative positions, which would lead to 9 individual states. The way we have defined the game is to condense the state into a single one which has same normal form. This is done by considering the independence of the movement in the horizontal direction. The only difference between the way in which we have defined is the transition of the agents into different states this happens because of occurrence of invalid state in the conventional game representation.

3.0.6 Utility

The game is defined as Zero sum Game, and hence one player loss is the other player's gain.The payoff matrix is defined for each state.

Since the game is a zero sum game,

$$u_{\bigtriangleup}(action_1, action_2) = -u_{\bigtriangledown}(action_1, action_2)$$

The normal form representation is given the next section. Values of the payoff is based on the action each player selects.

3.0.7 Normal form representation of the states

The Normal form shows the payoff received.As this is a zero sum game the loss of one player is gain of other player. Energy or cost required by the player to play block is less as compared to attack, whereas there is no cost for the movement.

$P_\triangle$ is the first player, and $P_\triangledown$ is the second player. However this does not imply the game is played in sequence. The game is simultaneous game.Declaring the payoff of the other player is not required since its a zero sum game. They can be omitted or shown as follows:

$P_\triangle$ \ $P_\triangledown$	R	L	B	A
R	$(0,0)$	$(0,0)$	$(\frac{1}{2},-\frac{1}{2})$	$(1,-1)$
L	$(0,0)$	$(0,0)$	$(\frac{1}{2},-\frac{1}{2})$	$(1,-1)$
B	$(-\frac{1}{2},\frac{1}{2})$	$(-\frac{1}{2},\frac{1}{2})$	$(0,0)$	$(-\frac{1}{2},\frac{1}{2})$
A	$(-1,1)$	$(-1,1)$	$(\frac{1}{2},-\frac{1}{2})$	$(0,0)$

Table 3.1: 4x4 Matrix: s_0 Normal-Form Game

$P_\triangle$ \ $P_\triangledown$	R	L	B	A
R	$(0,0)$	$(0,0)$	$(\frac{1}{2},-\frac{1}{2})$	$(-1,1)$
L	$(0,0)$	$(0,0)$	$(\frac{1}{2},-\frac{1}{2})$	$(1,-1)$
B	$(-\frac{1}{2},\frac{1}{2})$	$(-\frac{1}{2},\frac{1}{2})$	$(0,0)$	$(0,0)$
A	$(-1,1)$	$(1,-1)$	$(0,0)$	$(0,0)$

Table 3.2: 4x4 Matrix: s_1 Normal-Form Game

$P_\triangle$ \ $P_\triangledown$	R	L	B	A
R	$(0,0)$	$(0,0)$	$(\frac{1}{2},-\frac{1}{2})$	$(1,-1)$
L	$(0,0)$	$(0,0)$	$(\frac{1}{2},-\frac{1}{2})$	$(-1,1)$
B	$(-\frac{1}{2},\frac{1}{2})$	$(-\frac{1}{2},\frac{1}{2})$	$(0,0)$	$(0,0)$
A	$(1,-1)$	$(-1,1)$	$(0,0)$	$(0,0)$

Table 3.3: 4x4 Matrix: s_2 Normal-Form Game

3.0.8 Transition of states based on actions taken

The actions taken in one state leads to the player to move in a different state this can be shown in the following transition table. If both the player take same action then they end up reaching same state.

State S_3 is the terminal state where the game ends. The game can start from any of the initial state S_0,S_1 and S_2. However it won't start from state S_3.

Transition table for Game I				
△	▽	From State S_0	From State S_1	From State S_2
R	R	S_0	S_1	S_2
L	R	S_2	S_0	S_1
B	R	S_1	S_2	S_0
A	R	S_1	S_3	S_0
R	L	S_1	S_2	S_0
L	L	S_0	S_1	S_2
B	L	S_2	S_0	S_1
A	L	S_3	S_0	S_1
R	B	S_2	S_0	S_1
L	B	S_1	S_2	S_0
B	B	S_0	S_1	S_2
A	B	S_0	S_1	S_2
R	A	S_3	S_0	S_1
L	A	S_1	S_3	S_1
B	A	S_0	S_1	S_2
A	A	S_3	S_3	S_3

The above transition table provides environment restriction while a game is being played.

3.1 Game theoretic solution for the Attack Game

State s_0 Player ▽ / Player △	R	L	B	A
R	$(0,0)$	$(0,0)$	$(0.5,-0.5)$	$(1,-1)$
L	$(0,0)$	$(0,0)$	$(0.5,-0.5)$	$(1,-1)$
B	$(-0.5,0.5)$	$(-0.5,0.5)$	$(0,0)$	$(-0.5,0.5)$
A	$(-1,1)$	$(-1,1)$	$(0.5,-0.5)$	$(0,0)$

Figure 3.2: There are multiple Nash equilibriums (R,R) , (R,L), (L,R),(L,L)

State s_1 Player ▽ / Player △	R	L	B	A
R	$(0,0)$	$(0,0)$	$(0.5,-0.5)$	$(-1,1)$
L	$\boxed{(0,0)}$	$(0,0)$	$(0.5,-0.5)$	$(1,-1)$
B	$(-0.5,0.5)$	$(-0.5,0.5)$	$(0,0)$	$(0,0)$
A	$(-1,1)$	$(1,-1)$	$(0,0)$	$(0,0)$

Figure 3.3: pure stage game S_1 equilibrium (L,R)

State s_2 Player $\bigtriangledown$ / Player $\bigtriangleup$	R	L	B	A
R	$(0,0)$	$(0,0)$	$(0.5,-0.5)$	$(1,-1)$
L	$(0,0)$	$(0,0)$	$(0.5,-0.5)$	$(-1,1)$
B	$(-0.5,0.5)$	$(-0.5,0.5)$	$(0,0)$	$(0,0)$
A	$(1,-1)$	$(-1,1)$	$(0,0)$	$(0,0)$

Figure 3.4: pure stage game S_2 equilibrium (R,L)

Nash equilibrium

1. There are multiple Nash equilibria for state s_0 (R,R) , (R,L), (L,R) & (L,L),
2. Nash equilibrium for state s_1 is (L,R)
3. Nash equilibrium for state s_2 is (R,L).

The above solution for individual stage provides an interesting incites to the problem. In the first stage of the game the player is indifferent between the R and L. This means the player will play the game with a range of probabilities. All the range of probabilities over R and L are Nash equilibrium.

For State S_1, from the perspective of player $\bigtriangleup$, the player is certain about playing action L. This is because Player receives more incentives to play L rather than R. This is due to the presence of negative reward if the player $\bigtriangledown$ chooses move A.

Similar explanation stands for player $\bigtriangleup$ but in the reverse way. If one observes the payoff received by player playing R gets player more incentive as compared to R. This is also due to the presence of negative rewards corresponding to A move of other player if player $\bigtriangleup$ selects L.

Mixed strategy

It will be interesting to see what mixed strategy players will apply while playing the game. The following is the description how the player will select a mixed strategy for state S_0 and State S_1. The same concept can be extended to the state S_2.

State S_0

$$\begin{array}{cc|c|c|} & & \multicolumn{2}{c}{P_{\bigtriangledown}} \\ & & R & L \\ \cline{3-4} \multirow{2}{*}{$P_{\bigtriangleup}$} & R & 0 & 0 \\ \cline{3-4} & L & 0 & 0 \\ \cline{3-4} \end{array}$$

Table 3.4: 4x4 Matrix: s_0 Reduced Normal-Form Game by Iterative deletion of dominant strategy

Consider the normal form of state S_0, it can be seen that the payoff recieved by player $\triangle$ for R and L are same and at the same time player $\triangle$ is better off playing R and L as compared to B and A.The normal form can be reduced using applying iterative deletion of dominant strategy. The strategy which won't be ever is removed and what is left, player mixes with those actions. The reduced Normal form will be as shown above.

In the reduced Normal form, we can see player is indifferent to the action. Hence the following is the range where probabilities lies

$$P_R \in [0, 1]$$

$$P_L \in [0, 1]$$

under the constraints,

$$P_R + P_L = 1$$

State S_1

It can be seen that for state S_2 by Iterative deletion of dominant strategy, the normal form is reduced to the following form The expected payoff for the player $\triangle$ is shown as follows

		$P_\triangledown$		
		R	L	A
	R	0	0	-1
$P_\triangle$	L	0	0	1
	A	-1	1	0

Table 3.5: 4x4 Matrix: s_1 Reduced Normal-Form Game by Iterative deletion of dominant strategy

$$E_\triangle(R) = -1 + p + q$$

$$E_\triangle(L) = 1 - p - q$$

$$E_\triangle(A) = -p + q$$

Equating, $E_\triangle(R) = E_\triangle(L)$

$$-1 + p + q = 1 - p - q$$

$$p + q = 1$$

This shows

$$P_A = 0$$

The negation will be expected payoff recieved by $\triangledown$ when it plays probability p and q.

p	q	Payoff for $P_\bigtriangledown$	Payoff for $P_\bigtriangledown$	Payoff for $P_\bigtriangledown$
P_R	P_L	$P_\triangle \rightarrow$ R	$P_\triangle \rightarrow L$	$P_\triangle \rightarrow A$
0	1	0	0	-1
1	0	0	0	1
0.5	0.5	0	0	0
0.25	0.75	0	0	-0.5
0.75	0.25	0	0	0.5
0.51	0.49	0	0	0.01

Table 3.6: Expected payoff for $P_\bigtriangledown$ for mixing different probabilities

The above table shows the expected payoff which it receives for the mixing P_R and P_L, This means

$$P_R \geq P_L$$

.The reason for this is player $P_\bigtriangledown$ receives atleast 0 or better than that for the probabilities. Mixed strategy will be as follows:

$$P_R \geq P_L$$

under constraints

$$P_R + P_L = 1$$

,

$$P_R \geq 0.5$$

$$P_L \leq 0.5$$

Similar explanation can be given for Player $P_\triangle$ and also for State S_2

4

Methodology

1. The above game can be seen as a optimisation problem in which one has to find a solution from a search space which is optimal. The solution will be a mixed strategy for sure (as per Nash's dissertation we always have a mixed strategy equilibrium for a game). One of the way to solve such a problem and find a solution is using Genetic Algorithm and Coevolutionary algorithm [12] [24]. Which are essentially same thing the only difference is in coevolutionary algorithm Multiple GA are run in parallel with interactions.

2. In each state of the game the player will play their action with certain probabilities[18]. The optimal strategy is defined by the strategies developed in each stage.

3. The following probabilities in this case will be

$$P_1(a/s_0), P_1(a/s_1), P_1(a/s_2)$$

$$P_2(a/s_0), P_2(a/s_1), P_2(a/s_2)$$

For the matrix game in state s_0 we will have a solution which will be

Generation	Optimal Value	$P_\triangle(R/s_0)$	$P_\triangle(L/s_0)$	$P_\triangle(B/s_0)$	$P_\triangle(A/s_0)$
G_x	value	p_1	p_2	p_3	p_4
G_y	value	q_1	q_2	q_3	q_4

Table 4.1: solution for $P_\triangle\ s_0$

The above table consists of (p,q) which are probabilities under constraint

$$p_1 + p_2 + p_3 + p_4 = 1$$

$$q_1 + q_2 + q_3 + q_4 = 1$$

4.1 Co-Evolutionary Algorithm

Genetic Algorithm

Genetic algorithm mimics the concept of natural evolution. In an optimisation problems it reaches to a stable solution which is close to or the optimal point in search. Genetic algorithm basic structure which has been used in the research can be explained as

However, the above structure is useful to identify strategy for either of the player.[12] [8]. Instead of

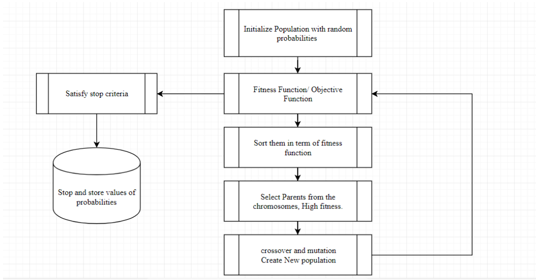


Figure 4.1: Genetic Algorithm Outline

running multiple instances of GA one after the another we can run them in parallel this can be done using Co-Evolutionary algorithm [22]. Each instance of evolve into a species of individuals which are good at particular tasks.

1. Gene : Each gene represents the value of probability the player assigns to the actions.
2. Evolution of the population : During evolution stage each individual is matched against each other individuals for a fixed number of time.
3. Selection : Based on the result of the game the best players are selected to mate and produce offspring.
4. Crossover : The selected parents are allowed to reproduce to new generation.
5. Mutation : The selected chromosome is mutated one bit mutation.
6. New generation: New generation is selected from the pool of children depending on the fitness of the generated offspring.

The game which we are looking at is a purely competitive game, and with multiple stages while searching for solution in such one may encounter more than one optimal solution. These solutions are optimal when all objectives are into consideration and are Paretooptimal solutions [35]

Multiobjective Optimization Problem:

We have multiple objective functions each for individual states this objective functions are aggregated to form a composite function which can be optimised at once.

Genetic algorithm mimics the concept of natural evolution. In an optimisation problems it reaches to a stable solution which is close to or the optimal point in search. Genetic algorithm basic structure which has been used in the research can be explained as

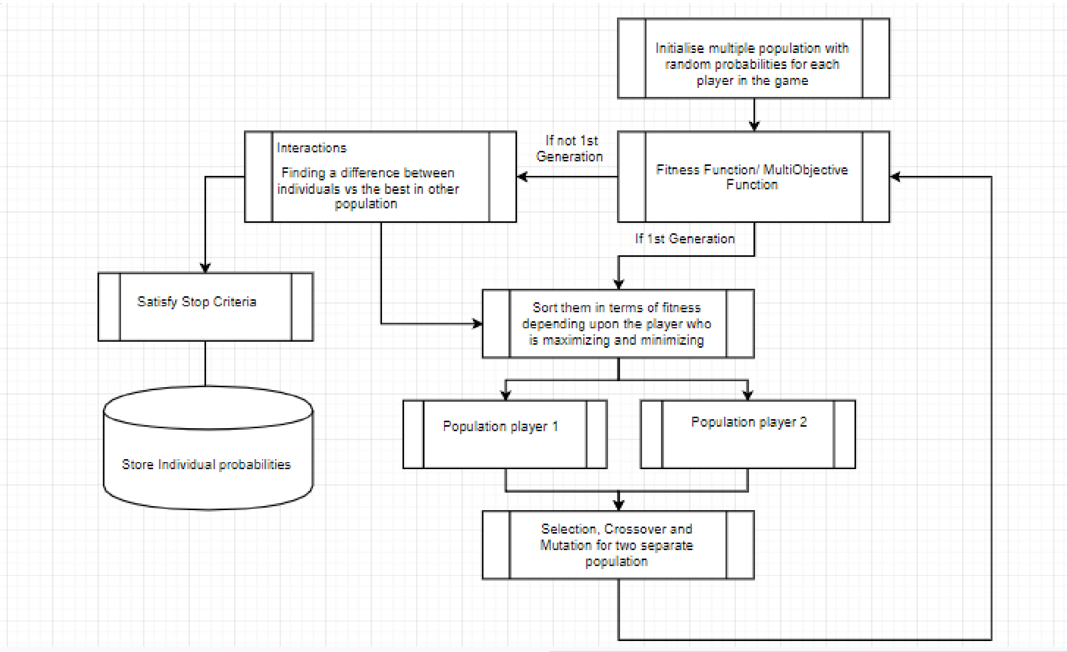


Figure 4.2: Coevolutionary Algorithm Outline

4.2 Solving s_0 as an example by Hand

Based on [25].

4.2.1 Population and Gene

1. The above game is a zero sum game, one player's loss is other player's gain.
2. The player does not require to have any memory to play against each other. So the representation does not require the encoding of the memory as it is required in other games discussed in literature.
3. The player plays with certain mixed strategy. (Nash: Every game have a mixed strategy equilibrium)
4. This strategies are evolved for the players. The probabilities are evolved over time which may converge within the optimal strategy.

The initial example population

Gene	$P_1(L/s_0)$	$P_1(R/s_0)$	$P_1(A/s_0)$	$P_1(L/s_0)$
1	0.15	0.15	0.4	0.30
2	0.2	0.4	0.2	0.2
3	0.1	0.1	0.6	0.7
4	0.65	0.05	0.05	0.25
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.

Table 4.2: Gene example for $P_{\triangle}\ s_0$ population

This above population and encoding.[17] An example of encoding a value of the probability to be encoded is as follows: 0.5 to be encoded.

Population size: N = 24

The length of the string of each parameter = 8

The equation used for this is as follows:

$$P_1 = P_{1min} + \frac{b}{2^{p_L}}(P_{1max} - P_{1min})$$

So for the above example it will be

$$0.5 = 0 - \frac{b}{2^8 - 1}(1 - 0)$$

binary representation is given by:

$$b = 0.5(2^8 - 1)$$

Now this decimal number can be converted into binary

$$B = dec2bin(127.5000, 8) =' 01111111'$$

4.2.2 Fitness Function

The implementation is being developed based on example [12] . Consider a zero sum game with m x n matrix.

For such a game Player 1 optimum mixed strategy will satisfy

$$Max[Min(\sum_{i=1}^{m} a_{i1}p_1(a_{i1}/s_0), \sum_{i=1}^{m} a_{i2}p_1(a_{i2}/s_0), ..., \sum_{i=1}^{m} a_{in}p_1(a_{in}/s_0))]$$

Subject to the constraints

$p_1(a_{i1}/s_0) \geq 0$, i=1,,m and $\sum_{i=1}^{m} p_1(a_{i1}/s_0) = 1$

A similar objective function can be defined for player 2:

$$Min[Max(\sum_{i=1}^{n} a_{i1}p_2(a_{i1}/s_0), \sum_{i=1}^{n} a_{i2}p_2(a_{i2}/s_0), ..., \sum_{i=1}^{n} a_{in}p_2(a_{im}/s_0))]$$

Subject to the constraints

$p_2(a_{i1}/s_0) \geq 0$, i=1,,n and $\sum_{i=1}^{n} p_2(a_{i1}/s_0) = 1$

An Example for the fitness value calculation is Say the example probability are

Gene	$P_1(L/s_0)$	$P_1(R/s_0)$	$P_1(A/s_0)$	$P_1(L/s_0)$
decimal	0.4	0.2	0.2	0.2
Binary encoding	01100110	00110011	00110011	00110011

Table 4.3: $P_\triangle$ s_0 example and corresponding binary encoding

s_0 $P_\triangledown$ / $P_\triangle$	R	L	B	A
R	0	0	0.5	1
L	0	0	0.5	1
B	-0.5	-0.5	0	-0.5
A	-1	-1	0.5	0

Table 4.4: Game Matrix in state s_0

Simplifying the objective function for a single person and in terms of the game can be written as

$$objValue = Max\left[Min\begin{bmatrix} 0*P_1(L/s_0)+0*P_1(R/s_0)+(-0.5)*P_1(B/s_0)+(-1)*P_1(A/s_0) \\ 0*P_1(L/s_0)+0*P_1(R/s_0)+(-0.5)*P_1(B/s_0)+(-1)*P_1(A/s_0) \\ 0.5*P_1(L/s_0)+0.5*P_1(R/s_0)+0*P_1(B/s_0)+0.5*P_1(A/s_0) \\ 1*P_1(L/s_0)+1*P_1(R/s_0)+(-0.5)*P_1(B/s_0)+0*P_1(A/s_0) \end{bmatrix}\right]$$

$$= Max\left[Min\begin{bmatrix} 0*0.4+0*0.2+(-0.5)*0.2+(-1)*0.2 \\ 0*0.4+0*0.2+(-0.5)*0.2+(-1)*0.2 \\ 0.5*0.4+0.5*0.2+0*0.2+0.5*0.2 \\ 1*0.4+1*0.2+(-0.5)*0.2+0*0.2 \end{bmatrix}\right] = Max\left[Min\begin{bmatrix} -0.3000 \\ -0.3000 \\ 0.4000 \\ 0.5000 \end{bmatrix}\right]$$

$$= Max\left[Min\begin{bmatrix} -0.3000 \\ -0.3000 \\ 0.4000 \\ 0.5000 \end{bmatrix}\right]$$

$$= -0.300$$

Gene	$P_1(L/s_0)$	$P_1(R/s_0)$	$P_1(A/s_0)$	$P_1(L/s_0)$	Objective Function
1	0.4	0.2	0.2	0.2	-0.30
2	0.1	0.1	0.2	0.6	-0.70
3	0.4	0.4	0.1	0.1	-0.15

Table 4.5: fitness function value $P_\triangle$ s_0 population

Gene	$P_1(L/s_0)$	$P_1(R/s_0)$	$P_1(A/s_0)$	$P_1(L/s_0)$
decimal	0.4	0.2	0.2	0.2
Binary encoding	01100110	00110011	00110011	00110011
decimal	0.1	0.1	0.2	0.6
Binary encoding	00011001	00011001	00110011	10011001
decimal	0.4	0.4	0.1	0.1
Binary encoding	01100110	01100110	00011001	00011001

Table 4.6: $P_{\triangle}\ s_0$ population with binary encoded values

Chromosome	fitness value
01100110 | 00110011 | 00110011 | 00110011	-0.3
00011001 | 00011001 | 00110011 | 10011001	-0.7
01100110 | 01100110 | 00011001 | 00011001	-0.15

Table 4.7: $P_{\triangle}\ s_0$ binary encoded population

4.2.3 Selection:

Selection is the process to select the individual from the population to reproduce. One can identify the fittest from the population and select them to mate however this may lead to a wrong convergence. In order to ensure we avoid this, one needs to maintain diversity in the selection process.

Here we use Tournament Selection

1. The fitness values are normalised

The normalised values are as follows:

Chromosome	fitness value	Normalised value
01100110 | 00110011 | 00110011 | 00110011	-0.3	0.2609
00011001 | 00011001 | 00110011 | 10011001	-0.7	0.6087
01100110 | 01100110 | 00011001 | 00011001	-0.15	0.1304

Table 4.8: $P_{\triangle}\ s_0$ population with fitness and noramlised values

Sorted list:

Chromosome	fitness value	Normalised value
00011001 | 00011001 | 00110011 | 10011001	-0.7	0.6087
01100110 | 00110011 | 00110011 | 00110011	-0.3	0.2609
01100110 | 01100110 | 00011001 | 00011001	-0.15	0.1304

Table 4.9: $P_{\triangle}\ s_0$ population

Let a number is selected at random, R = 0.2 The chromosome greater or equal to 0.2 is selected.

Selected	Chromosome
parent 1	00011001 | 00011001 | 00110011 | 10011001
parent 2	01100110 | 00110011 | 00110011 | 00110011

Table 4.10: $P_{\triangle}\ s_0$ sorted population with respect to fitness

4.2.4 Crossover:

Crossover point is selected at random: Let say it happens at 14

Selected	Chromosome
parent 1	00011001 | 000110-01 | 00110011 | 10011001
parent 2	01100110 | 001100-11 | 00110011 | 00110011

Table 4.11: Crossover: selected parents

Crossover example:

Selected	Chromosome
child 1	00011001 | 000110-11 | 00110011 | 00110011
child 2	01100110 | 001100-01 | 00110011 | 10011001

Table 4.12: Crossover childrens

The problem here is the selection with the child or the offspring as we can see the children are as follows

Selected	Chromosome	total
child 1	0.098 | 0.1059 | 0.2 | 0.2	0.6039
child 2	0.4 | 0.1922 | 0.2 | 0.6	1.3922

Table 4.13: Decoded Childrens from crossover

The genome being a distribution of probability on the set of actions, We must need to add constraints to crossover and mutation operator.

The following are the constraints that must be satisfied

N = 4, support : i $\subset$ (1,2,3,4)

$0 \leq P_i \leq 1$

Thus,crossover and mutations have to be performed carefully in order to generate individuals with genomes. It is not difficult to observe that all the crossover operators commonly used in the literature can be used in this case if followed by normalisation step to correct the genome to have all the genes summing up to 1.

The normalisation also introduces an additional mutation component because the generated genome does not necessarily contains the same genes as the parents.

Normalised representation:

Selected	Chromosome	total
child 1	0.1623 | 0.1754 | 0.3312 | 0.3312	1.000
binary	00101001 | 00101100 | 01010100 | 01010100	11111111
child 2	0.2873 | 0.1381 | 0.1437 | 0.4310	1.000
binary	01001001 | 00100011 | 00100100 | 01101101	11111111

Table 4.14: $P_{\triangle}$ normalised

4.2.5 Mutation:

A single bit mutation is being added to the selected individuals.

4.2.6 Fitness Evaluation:

The process repeats for the number of plays in each iteration for both the players. The fitness is not only evaluated for individual they are also evaluated based on the other population. This is done on by matching the population with the best player of the other population.

New population is created then this process is repeated till the algorithm converges to an expected fitness or the number of maximum generations are over.

5

Simulation

5.1 Simulation for Individual States using CoEv

The Game defined is a multistage zero sum game, however the individual stages can be considered as a distinct problem and solved for. If considered separately, Individual stages can be looked at as a single objective optimisation problems, which can be solved using Coevolutionary algorithm. This simulation performs individual stage optimisation in order to find optimal mixed strategy equilibria. Solution is compared with the Nash equilibrium obtained in order to establish relevance of the algorithm. Further, This simulation allows us to identify.

1. *How an agent or a player would select the mixed strategy if evolved for playing individual stages?*
2. *Does this Coevolution corresponds to the Nash equilibrium strategies obtained earlier?*
3. *Can a solution for the individual be considered as optimal solution?*

5.1.1 Individual Stage Game S_0 simulation

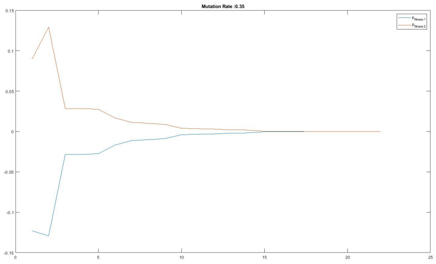
Parameters	Values
N	48
C_R	0.35
M_R	0.4
I	96

(a) Parameters

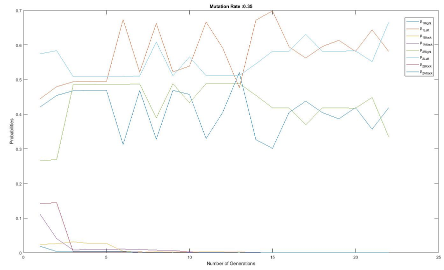
State	Player	P_{Right}	P_{Left}	P_{Block}	P_{Attack}	Gx
S-I						
s_0	$P_\triangle$	0.4188	0.5812	0	0	22
s_0	$P_\bigtriangledown$	0.3345	0.6655	0	0	22
S-II						
s_0	$P_\triangle$	0.3494	0.6506	0	0	28
s_0	$P_\bigtriangledown$	0.446	0.554	0	0	28

(b) Probabilities observed through simulations

Table 5.1: Parameters used for simulation S_0 and converged probabilities

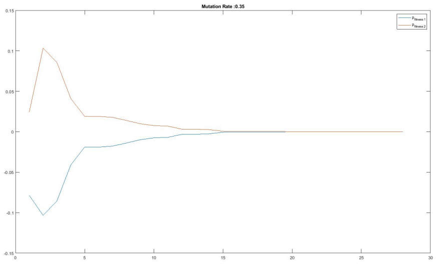


(a) Fitness

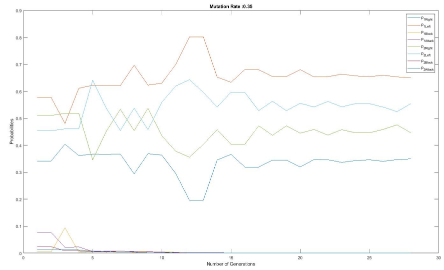


(b) Converged Probabilities

Figure 5.1: State S_0 simulation I



(a) Fitness



(b) Converged Probabilities

Figure 5.2: State S_0 simulation II

The simulation

5.1.2 Individual Stage Game S_1 simulation

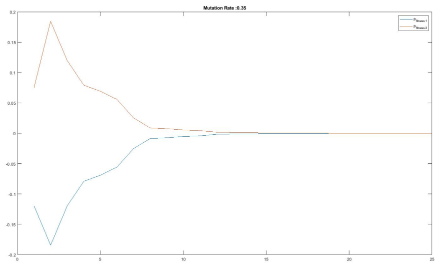
Parameters	Values
N	48
C_R	0.35
M_R	0.4
I	96

(a) Parameters

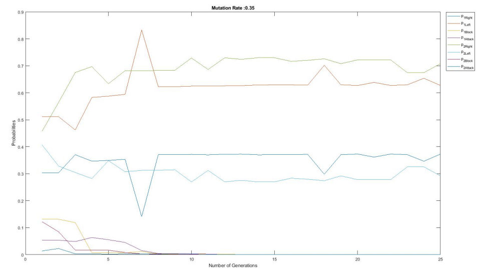
State	Player	P_{Right}	P_{Left}	P_{Block}	P_{Attack}	Gx
S-I						
s_0	$P_{\triangle}$	0.3729	0.6271	0	0	25
s_0	$P_{\bigtriangledown}$	0.708	0.292	0	0	25
S-II						
s_0	$P_{\triangle}$	0.2767	0.7233	0	0	24
s_0	$P_{\bigtriangledown}$	0.6481	0.3519	0	0	24

(b) Probabilities observed through simulations

Table 5.2: Parameters used for simulation S_1 and converged probabilities

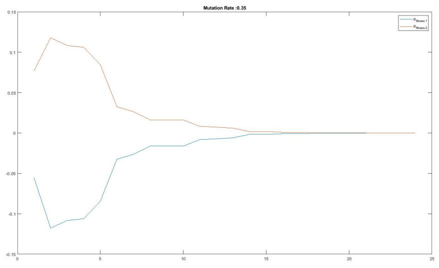


(a) Fitness

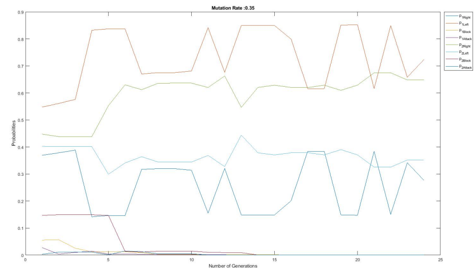


(b) Converged Probabilities

Figure 5.3: State S_1 simulation I



(a) Fitness



(b) Converged Probabilities

Figure 5.4: State S_1 simulation II

5.1.3 Individual Stage Game S_2 simulation

Simulation for state S_2

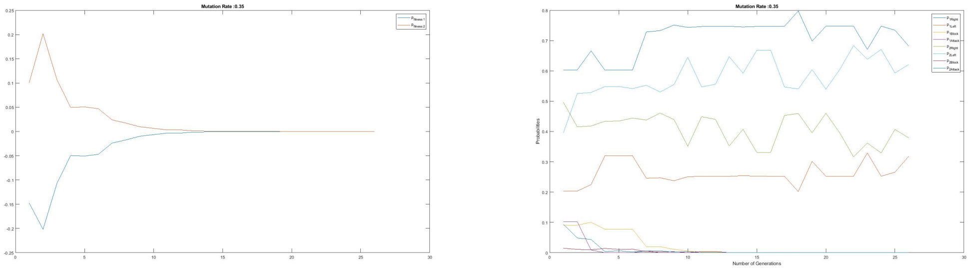
Parameters	Values
N	48
C_R	0.35
M_R	0.4
I	96

(a) Parameters

State	Player	P_{Right}	P_{Left}	P_{Block}	P_{Attack}	Gx
S-I						
s_0	$P_{\triangle}$	0.6815	0.3185	0	0	26
s_0	$P_{\triangledown}$	0.3787	0.6213	0	0	26
S-II						
s_0	$P_{\triangle}$	0.6529	0.3471	0	0	28
s_0	$P_{\triangledown}$	0.4193	0.5807	0	0	28

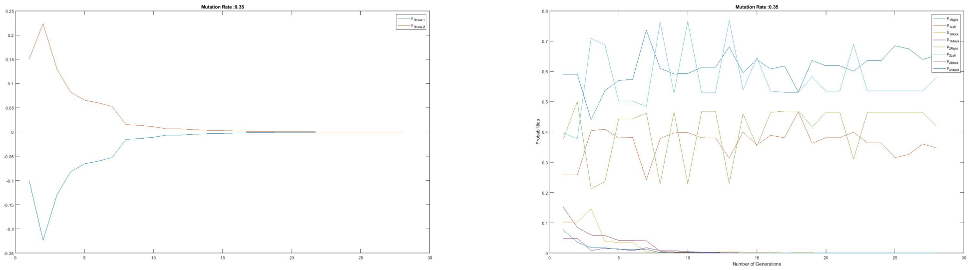
(b) Probabilities observed through simulations

Table 5.3: Parameters used for simulation S_2 and converged probabilities



(a) Fitness (b) Converged Probabilities

Figure 5.5: State S_2 simulation I



(a) Fitness (b) Converged Probabilities

Figure 5.6: State S_2 simulation II

5.1.4 Observation

Multiple crossover rate and mutation rate

State	Player	P_{Right}	P_{Left}	P_{Block}	P_{Attack}
s_0	$P_{\triangle}$	$0 \leq P_R \leq 1$	$0 \leq P_L \leq 1$	$P_B = 0$	$P_A = 0$
s_1	$P_{\triangle}$	$0 \leq P_R \leq 0.5$	$0.5 \leq P_L \leq 1$	$P_B = 0$	$P_A = 0$
s_2	$P_{\triangle}$	$0.5 \leq P_R \leq 1$	$0 \leq P_L \leq 0.5$	$P_B = 0$	$P_A = 0$
s_0	$P_{\bigtriangledown}$	$0 \leq P_R \leq 1$	$0 \leq P_L \leq 1$	$P_B = 0$	$P_A = 0$
s_1	$P_{\bigtriangledown}$	$0.5 \leq P_R \leq 1$	$0 \leq P_L \leq 0.5$	$P_B = 0$	$P_A = 0$
s_2	$P_{\bigtriangledown}$	$0 \leq P_R \leq 0.5$	$0.5 \leq P_L \leq 1$	$P_B = 0$	$P_A = 0$

Table 5.4: Probabilities observed through simulations

1. There is a large number of strategies as a solution for the Game.
2. Probabilities for Attack and block for all the stages and players is converging to zero which means players won't be benefited by playing those in the game.
3. Probabilities for moves Right and Left fluctuate between a range, indeterminate which one to select for sure, player appears to be benefited with all the probability combinations.

5.1.5 Analysis

As one can observe, Solution for individual states S_0, S_1, S_2 defined in section *"Game Definition"* of *"Game and Game Solution"* does not have a unique Nash equilibrium.

State	Nash equilibria: ($P_\triangle$,$P_\bigtriangledown$)
s_0	$(R, R), (R, L), (L, R), (L, L)$
s_1	(L, R)
s_2	(R, L)

Table 5.5: Nash equilibria

With a superficial observation, a changing outcome of table 5.1 may seem improbable but this is not true. One can clearly observe

1. According to the data obtained from the simulations one can observe that the problem of individual stage i.e single objective optimisation of stage is a problem with infinite solution set.

The above statements can further be supported by investigating pay-off matrix

1. Evolving Player, $P_\triangle$ for state S_0 receives same pay-off irrespective of move Right or Left. This makes player indifferent to this moves. This indifference is projected in the form of different probabilities convergence for different initial population of the players.

2. Evolving player $P_\triangle$, for state S_1, taps more probability on the Left as opposed to Right. This can be explained with the pay-off matrix: The pay-off matrix shows that player $P_\triangle$ receives same pay-off for action right and left if the player $P_\bigtriangledown$ plays right, left or block however if the player $P_\bigtriangledown$ plays Attack, Left is a better move for player $P_\triangle$ as it receives 1 in this case whereas right move causes a loss of 1. This is also translated into the range of probabilities $P_\triangle$ which is P_{right} in-between 0 and 0.5. i.e Right has a lower probabilities. This thought can further be cemented by considering how CoEv is implemented for individual stage. The player $P_\triangle$ is maximising whereas player $P_\bigtriangledown$ is minimizing. As in case of procedure to find Nash equilibrium. However, due to the presence of many optimal solution in the search space a saddle-point can be reached with different permutation and combination of probabilities. This is reason for different probabilities convergence.

3. A similar explanation holds true for other stages as well.

5.1.6 Conclusion for the simulation

CoEvolution mimics Nash equilibrium. Player $P_\triangle$ is maximising and player $P_\bigtriangledown$ is minimizing while evolving. The individual stages can be modelled as an optimisation problem however they are infinite solution set. There are many combination of probabilities of Right and Left which can form a saddle point and make the algorithm converge.Also, It is clear from the simulation that it is the rationality of the player not to select Moves Attack or block.

5.2 Simulations for Multistage Game using CoEv Algorithm

5.2.1 Simulation

Simulation explores the following idea

1. *What difference can one observe about players that are Co-Evolved for individual stage and multiple stages?*

2. *Does Co-Evolution for Multiple stages converges to a specific probability set instead of range of probabilities?*

3. *Why this happens?*

Multistage results

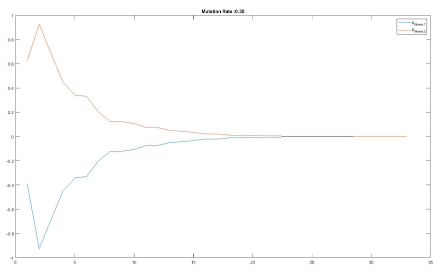
Parameters	Values
Gx	33
N	48
C_R	0.35
M_R	0.4
I	96

(a) Parameters

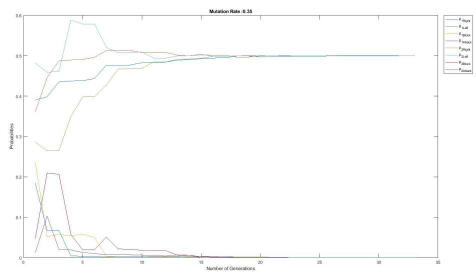
Player	P_{Right}	P_{Left}	P_{Block}	P_{Attack}
$P_{\triangle}$	0.4999	0.5001	0	0
P_{∇}	0.5	0.5	0	0

(b) Probabilities observed through simulations

Table 5.6: Parameters used for simulation and converged probabilities



(a) Fitness



(b) Converged Probabilities

Figure 5.7: State $S_0 + S_1 + S_2$ simulation I

5.2.2 Observations

1. Unlike Single objective Case, Multi-Objective Co-evolution converges to a specific probability point(0.5,0.5,0,0).

2. No matter how many times the process is repeated the solution is converged to same point.

5.2.3 Analysis

Table shows the range of probabilities for individual stages. This shows the valid probabilities that can occur in the individual stages. From the results we can observe, The mixed strategy is [0.5,0.5,0,0]. This is a valid probability for all the states.

1. The player tries to be good at all the three stages. This can be viewed as an intersection of the range of probabilities for all the stages.

2. The same player can play all the three games and entire game.

5.2.4 Conclusion for the simulation

Instead of multiple mixed strategies, when the game is treated as Multi-objective optimisation problem where objective function is converted into a composite function, a single mixed strategy is evolved. Convergence is a result of optimisation of all the function at the same time. Here, the converged mixed strategy is an intersection of solutions of all three stages of game. This makes evolved player to play in all the three stages with the same mixed strategy.

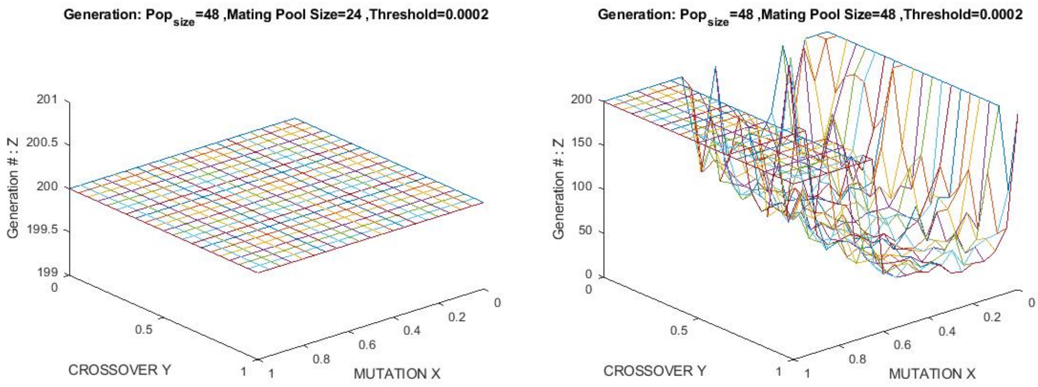
5.3 Simulation for Parameters with optimal Computation time

5.3.1 Simulation

A mixed strategy has been identified by the co-evolutionary algorithm in the above simulations. Here Computational speed or time has not been considered. Ideally, Computational time should be as small as possible. This simulation tries to do the following

1. *How Computational speed is affected by parameters involved in Co-Evolutionary Algorithm?*
2. *Can we identify a relationship between the parameter and Computational time?*
3. *Can parameters values be identified for optimal Computational speed?*

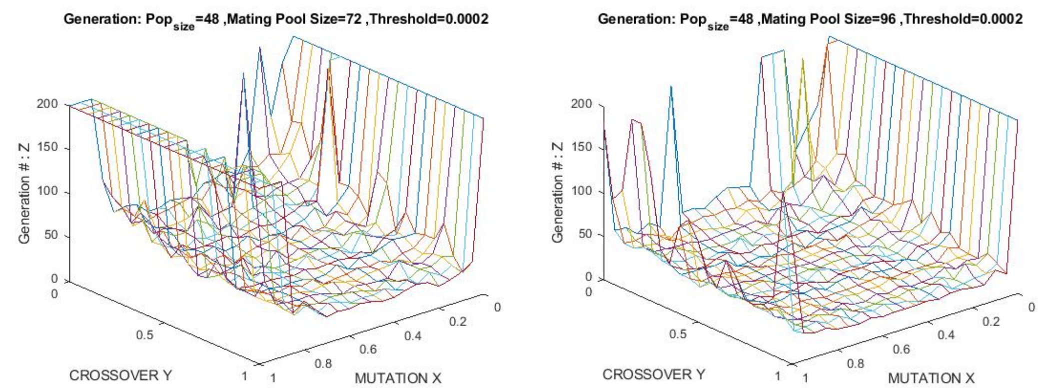
Changing mutation rate and crossover rate



(a) S-24

(b) S-48

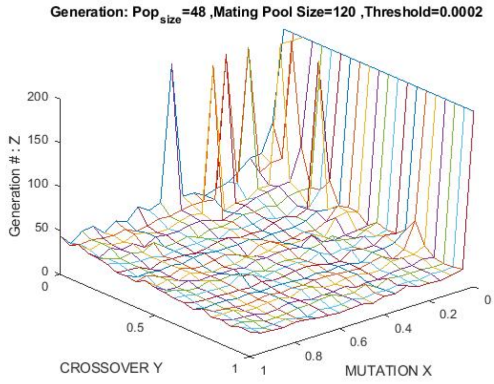
Figure 5.8: Interactions 24 and 48



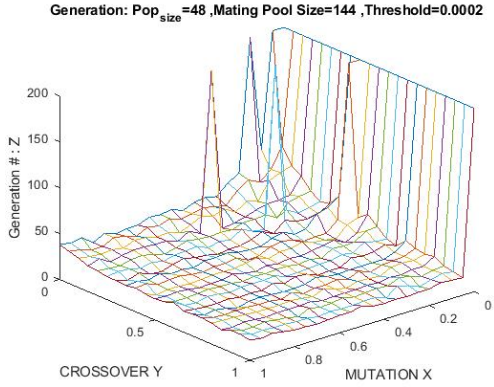
(a) S-72

(b) S-96

Figure 5.9: Interactions 72 and 96

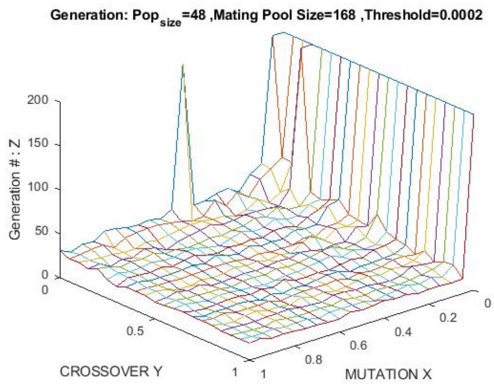


(a) S-120

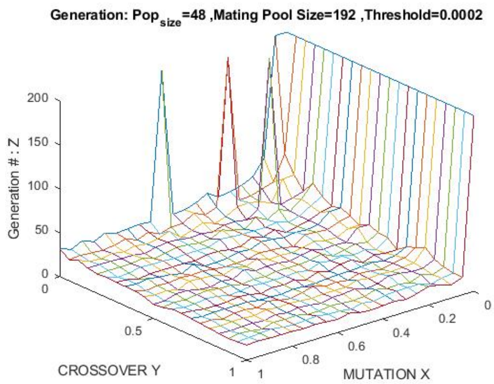


(b) S-144

Figure 5.10: Interactions 120 and 144

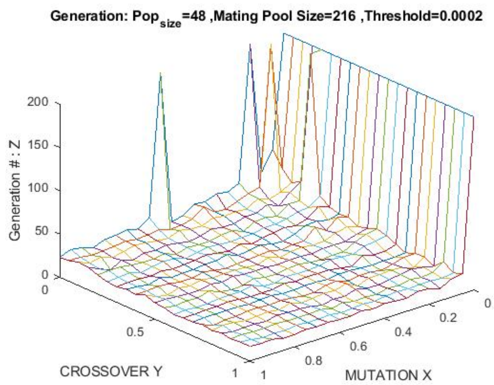


(a) Fitness

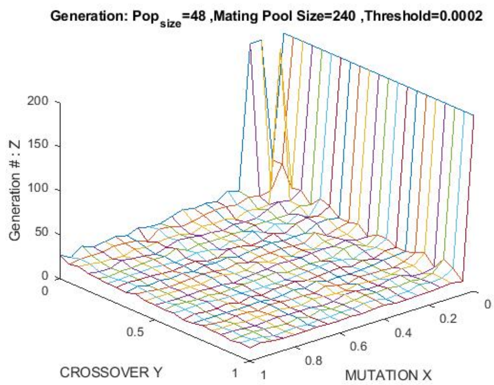


(b) S-192

Figure 5.11: Interactions 168 and 192



(a) S-216



(b) S-240

Figure 5.12: Interactions 216 and 240

5.3.2 Observation

1. A relationship between Generations and Interactions which is non-linear in nature can be observed.

2. The figures shows the change in the interactions with a constant population size of 48.

3. It can be seen that the increase in the interaction reduces the number of generations. Figures 5.8 and 5.9 shows for a mating pool size or interactions of more than 1.5 times of population size. The range over which convergence occur increases with the increase in interactions and number of generations took for convergence reduces.

4. Figure 5.10 to 5.12 shows the range over which convergence occur increases however, it unlike earlier figures Generations aren't that effected by change in interactions.

5. For a Mutation Rate of 0, No matter what the crossover rate and mating pool size is no convergence occurs.

5.3.3 Analysis

1. The reason for reduction in the number of generation with the interaction can be explained with the selection probability of an individual from the population. The equation is been explained in the next section "Derivation for computational time". As the Number of interactions increases. There is an exponential increase in the probability of an individual being selected.

2. This increase reflects in a better representation of population. This also results in participation of individual in mutation and crossover. Thus population goes through lot of diversity. Diversity results in exploration in the search space which results in lesser number of Generations. This is reflected in figure 5.10 to 5.12.

5.4 Derivation for Computational time

probability of an individual from the population is selected is given by

$$P_{individual} = \frac{1}{N}$$

Probability of selection for Crossover or mutation

1. Probability of selection of an individual from a population of N is

$$\frac{1}{N}$$

2. Probability of an individual not being selected in one interaction is

$$1 - \frac{1}{N}$$

3. Probability of an individual not being selected in two interaction is

$$(1 - \frac{1}{N}) * (1 - \frac{1}{N})$$

4. Probability of an individual not being selected in I interaction is

$$(1 - \frac{1}{N})^I$$

5. Probability of being selected atleast once in I interaction is

$$P_S = 1 - (1 - \frac{1}{N})^I$$

Diversity in Generations:

Diversity in the population is induced through Mutation rate and crossover rate which can be expressed as

$$Diversity = M_r + C_r$$

Probability that crossover occurs on the selected individual

probability of A such that B has happened

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

therefore

$$P(A \cap B) = P(B) * P(A|B)$$

P_{Cr} is C_r which is nothing but $P(A|B)$
Thus, Probability of crossover such that an individual is selected is given by

$$P_{Cr} * P_S$$

$$P(C_s) = (C_R) * (1 - (1 - \frac{1}{N})^I)$$

Probability that crossover occurs on the selected individual

Similarly, Mutation happen on the selected individual is given by

$$(M_r) * (1 - (1 - \frac{1}{N})^I)$$

Mutation and crossover occur independently of each other and Probability of two independent event is given by

$$P(A \cap B) = P(A) * P(B)$$

$$P(C_s \cap M_s) = (C_r) * (M_r) * (1 - (1 - \frac{1}{N})^I)^2$$

Probability of Crossover or Mutation or both on selected individual

Probability on two mutually exclusive event is given as follows

$$P(A \cup B \cup both) = P(A) + P(B) - P(A \cap B)$$

$$P(C_s \cup M_s \cup both) = (C_R * (1 - (1 - \frac{1}{N})^I) + (M_R * (1 - (1 - \frac{1}{N})^I) - (C_r * M_r * (1 - (1 - \frac{1}{N})^I)^2)$$

$$P(C_s \cup M_s \cup both) = (C_R + M_R) * (1 - (1 - \frac{1}{N})^I) - (C_r * M_r * (1 - (1 - \frac{1}{N})^I)^2)$$

The above probability is for single individual and its selection probability for mutation and crossover.

Total time

Total time taken per generation for mutation and crossover over the entire population,

$$T = (t_M * M_R + t_C * C_R + t_N c * (1 - C_R) + t_{Mis}) * I$$

where,

t_M = time taken for a single mutation.

t_C = time taken for a single crossover.

t_{NC}= time taken for single non-crossover case.

t_{Mis} = time taken for miscellaneous process.

Equation:

Total time will also include Fitness computation time T_F

$$T_{total} = T + T_F$$

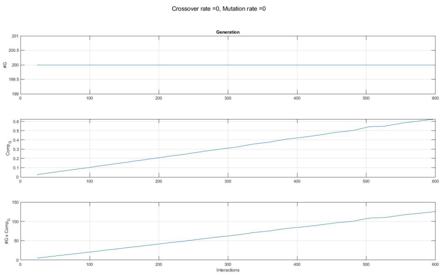
Total time taken G Generations,

$$T_G = G * T_{Total}$$

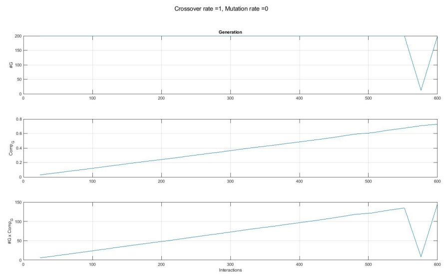
5.4.1 Relation between Generation, Interaction and Time taken

Generation & Interactions

The figure below shows the computational time required by using different Mutation rate and Crossover rate over a range of Interaction range [24,600].

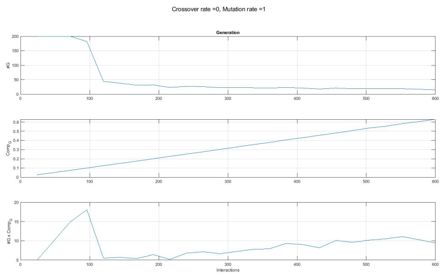


(a) $M_R = 0, C_R = 0$

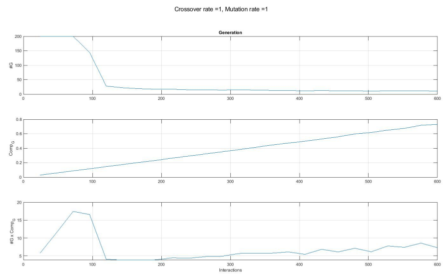


(b) $M_R = 0, C_R = 1$

Figure 5.13: Plot for G Vs I, Computation time Vs I and Total time Vs I

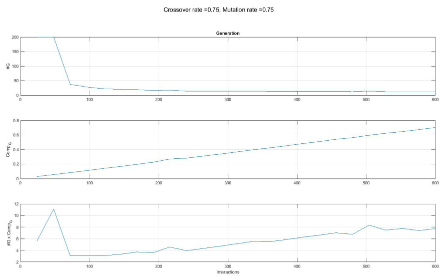


(a) $M_R = 1, C_R = 0$

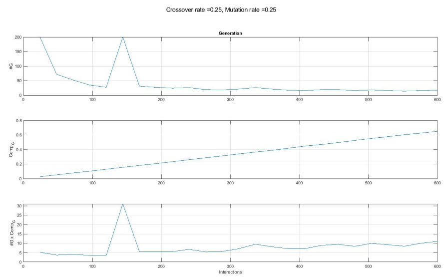


(b) $M_R = 1, C_R = 1$

Figure 5.14: Plot for G Vs I, Computation time Vs I and Total time Vs I

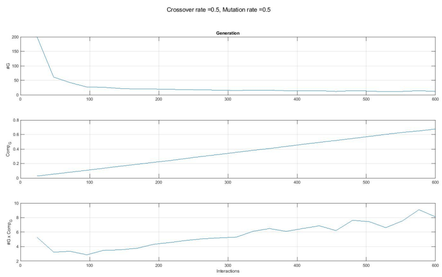


(a) $M_R = 0.75, C_R = 0.75$

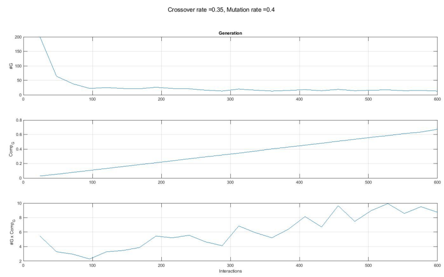


(b) $M_R = 0.25, C_R = 0.25$

Figure 5.15: Plot for G Vs I, Computation time Vs I and Total time Vs I



(a) $M_R = 0.5, C_R = 0.5$



(b) $M_R = 0.4, C_R = 0.35$

Figure 5.16: Plot for G Vs I, Computation time Vs I and Total time Vs I

Observations

1. First the computation time taken for crossover and mutation of an individual is constant through out the process. Again the population size is constant which also leads to a consistent fitness calculation time.

2. As it can be seen from the equation for computation time, Mutation rate and crossover rate effects the total computational time.

3. This can be verified from the figures above which shows

   (a) Fig. 5.13 shows for a $M_R = C_R = 0$, takes a long computation time which is obvious since there is no convergence

   (b) Fig 5.13 also shows a $M_R = 1$ and $C_R = 0$, for a smaller size of Interactions, No convergence occurs. However convergence starts from interaction size of 96. This is also the point where computation time is the least.

   (c) Fig. 5.14 shows for a $M_R = 0, C_R = 1$ no convergence occur which is in accordance to the earlier finding.

   (d) Fig. 5.14 also shows for a $M_R = 1, C_R = 1$ the least amount of time taken is for an interaction size of 120

   (e) Fig 5.15 and 5.16 reiterate the finding but with a different M_R and C_R

   (f) Fig 5.16 gives a better representation which shows least amount of time taken is for an Interaction size of 96.

4. point around appears to be the sweet spot which requires least amount of time in all cases.

5. The least of amount of time taken by the simulations where in the case of $M_R = 0.4$and $C_R = 0.35$ which is voids the notion that a smaller crossover and smaller mutation will take less computation time.

6. An important pattern which is in all the figures, there is the gradual increase in the computation time after interaction size of 96.

Analysis

1. Convergence occurs depending on the mutation rate , crossover rate and interactions size for the coevolving population. From the observation of previous section, one can note that without mutation no convergence occurs. Since there is no convergence the algorithm will be executed till the maximum allowable generation. Thus it takes higher computation time.

2. For most of the simulation convergence of the population occurs quicker for a mating pool size of 96. which is twice the size of population and from the developed equation of selection probability, one can see the selection probability is around 0.86.

3. For higher interactions the selection probability increases

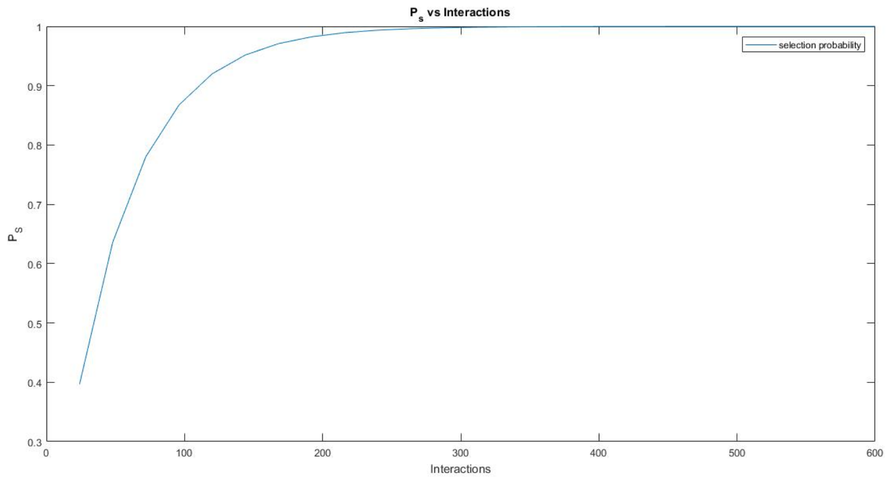


Figure 5.17: P_S vs I

4. The above plot shows that after interaction of 240 there is not much change in the selection probability.

5. Higher Mutation rate and crossover rate takes a bit longer time for convergence.However, Lower Mutation rate and crossover rate does not guarantee the least computational time. There is a range of Mutation rate and Crossover rate where computation time is less.

5.4.2 Comparison between values Actual time and real time taken for Interaction

Time is calculated from the equation derived in the derivation section. The derived time is plotted over actual time taken by the entire crossover, mutation and other involved process. This is shown in the figure below.

1. With the help of time for a single iteration an approximate time taken for the population to go through the mutation, crossover and other process can be observed which is then used for plotting.

2. The plotted time is for interactions 24 to 336 with a intermediate gap of 24 where x-axis is the combination of mutation rate and crossover rate. This is such that For a certain mutation rate Crossover rate changes from 0 to 1 with a gap of 0.05. There are 441 such combinations.

3. From the previous section, Interaction size of about 96 has been a good approximate for the convergence of the generations for various combination of mutation and crossover rate.

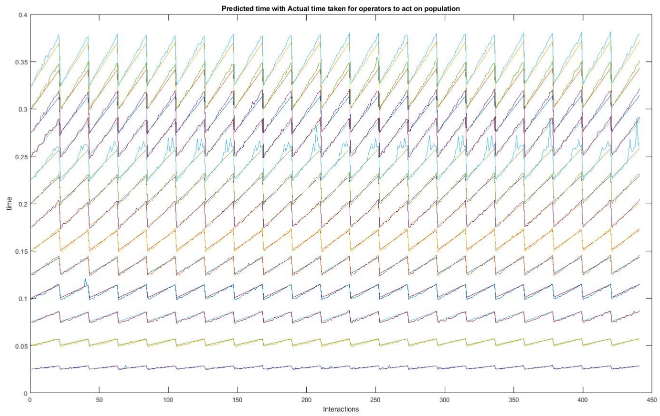


Figure 5.18: Game I States and their equivalent representation s_0,s_1,s_2

Observations and Analysis:

1. For higher interaction size the approximation has errors as compared to the actual time taken.
2. The time is calculated based on the processor of the machine it is running. Also, for most of the machines the time taken will vary based on the time allocated by the processor on the execution of the process.This causes a difference in the actual and calculated time.

5.5 Conclusion

There is no single point of mutation range and crossover rate for optimum computation time however a range which can be stated as Mutation rate $\in$ [0.2,0.5],Crossover rate $\in$ [0.2,0.6] and Interaction Size $\in$ [72,120]. From the range a point is selected for application to the new game which is as follows: This is because the time taken for the calculation is minimum for the simulation for different

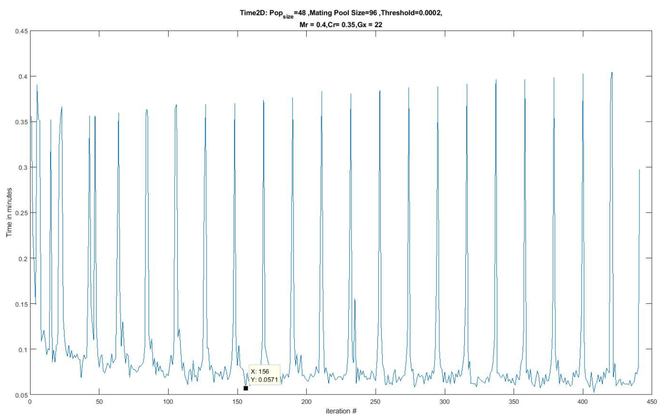


Figure 5.19: Game I States and their equivalent representation s_0,s_1,s_2

range of Mutation rate and Crossover rate. The selected point is $M_R = 0.4$,$C_R = 0.35$, and $I = 96$.

6

Game 2: Movement Game

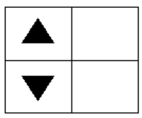
As opposed to the Horizontal Movement Game this game has been defined with not only horizontal but also with vertical movement. The horizontal and vertical movements takes the game a step close to the real world implementation. A deterministic game is the one in which if one takes certain actions he will reach to a unique result. We don't have any randomness in the process.

6.0.1 Players

Player $\triangle$and Player $\bigtriangledown$.

6.0.2 Environment

Zero sum Two player deterministic game is played in the following environment. There are four positions in the grid possible for the player to be in



Game II

6.0.3 Actions

Actions available for the players are

1. Right : Player moves to the right
2. Left : Player moves to the left
3. Up: Player moves up.
4. Down: Player moves down.
5. Block : Player tries to block the anticipated attack
6. Attack: Player Attacks.

6.0.4 States

Based on the players($\bigtriangledown$ and $\bigtriangleup$) and *Action set* (R,L,U,D,B,A) the number of states are defined as follows: The game consists of the following environment

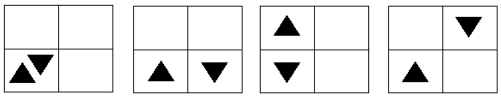


Figure 6.1: Game II States s_0,s_1,s_2,s_3

6.0.5 Utility

The game is defined as Zero sum Game, and hence one player loss is the other player's gain.The payoff matrix is defined for each state.

Since the game is a zero sum game,

$$u_{\bigtriangleup}(action_1, action_2) = -u_{\bigtriangledown}(action_1, action_2)$$

6.0.6 Representation choice

Conventionally, the states would have been defined in the terms of relative positions, which would lead to 16 individual states. The way we have defined the game is to condense the multiple state into a single one which has same normal form. The only difference between the way in which we have defined is the transition of the agents into different states this happens because of occurrence of invalid state in the conventional game representation. Unlike the previous game the agent is allowed to move in the vertical direction as well.

6.0.7 Normal form representation of the states

		$P_{\bigtriangledown}$			
		M_H	M_V	B	A
$P_{\bigtriangleup}$	M_H	0	0	0.5	1
	M_V	0	0	0.5	1
	B	-0.5	-0.5	0	-0.5
	A	-1	-1	0.5	0

Table 6.1: 4x4 Matrix: s_0 Normal-Form Game

		$P_{\bigtriangledown}$			
		M_H	M_V	B	A
$P_{\bigtriangleup}$	M_H	0	0	0.5	-1
	M_V	0	0	0.5	1
	B	-0.5	-0.5	0	0.5
	A	1	-1	-0.5	0

Table 6.2: 4x4 Matrix: s_1 Normal-Form Game

		$P_{\bigtriangledown}$			
		M_H	M_V	B	A
$P_{\bigtriangleup}$	M_H	0	0	0.5	1
	M_V	0	0	0.5	-1
	B	-0.5	-0.5	0	0.5
	A	-1	1	-0.5	0

Table 6.3: 4x4 Matrix: s_2 Normal-Form Game

		$P_{\bigtriangledown}$			
		M_H	M_V	B	A
$P_{\bigtriangleup}$	M_H	0	0	0.5	1
	M_V	0	0	0.5	1
	B	-0.5	-0.5	0	0.5
	A	-1	-1	-0.5	0

Table 6.4: 4x4 Matrix: s_3 Normal-Form Game

6.1 Simulation

1. Can Game I crossover and mutation range be used for Game II?

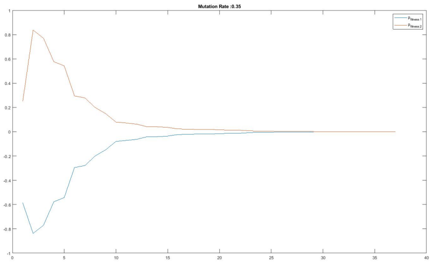
Parameters	Values
Gx	37
N	48
C_R	0.4
M_R	0.35
I	96

(a) Parameters

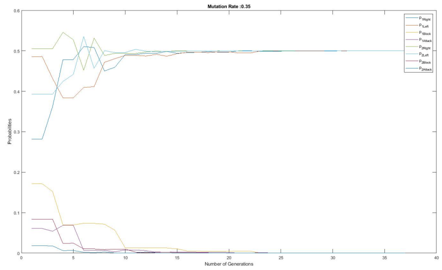
Player	P_{Right}	P_{Left}	P_{Block}	P_{Attack}
$P_{\triangle}$	0.5	0.5	0	0
$P_{\triangledown}$	0.5	0.5	0	0

(b) Probabilities observed through simulations

Table 6.5: Parameters used for simulation and converged probabilities



(a) Fitness



(b) Converged Probabilities

Figure 6.2: State $S_0 + S_1 + S_2 + S_3$ simulation I

6.2 Observation and Analysis

1. Convergence occurs with the given Mutation rate and crossover rate.
2. The converged values are [0.5,0.5,0,0].
3. Since the games are similar to each other only differing by single state, Convergence of the game is same as the previous game.

6.3 Conclusion

The selected mutation rate, crossover rate and interaction size developed from the earlier game can be applied to the new game. This proves the hypothesis that parameters can be used for convergence.

7

Discussion

7.1 Future Work

7.1.1 Multiple task Learning

The same methodology can be applied to Multiple task learning which are of similar nature. If one selects to apply Co evolutionary algorithm to the Multi task Learning problem. One can take into account similar approach which is applied in the thesis. If the tasks are considered as the stage of the game. The problem then can be reduced to essentially the same problem discussed in the research.

7.1.2 Thesis Extension

Research can be extended in numerous ways. Few of the extensions are discussed below in brief

Larger grid world

The problem can be extended to a larger grid world. This will be interesting to see how the agents behave in a world bigger than the current grid. The same representation can still hold good. However one may need to keep one more pointer to identify the edges of the game where action set will vary.

Interesting action set

The problem can be modelled with interesting set of actions apart from left,right,up, down,block and attack. This may also include diagonal movement.

More than two agents

It will be interesting to see how more than two agents behave in the given environment. The concept of cooperation may arrive in such a situation. That will be a more generalised solution to real world problems.

Reinforcement learning

A complete system can be created using reinforcement learning, where reinforcement learning can be used to learn unknown environment and the obtained rewards matrix can be used to generate a solution using CoEvolutionary algorithm.

7.1.3 Application of research

Direct application can be seen in development of two player game with an intelligent agent. The agent is evolved in reference to diverse population. This scenario can be applied to most of the real life situations such as bargaining, where one stage can be considered as first amount told, based on this outcome a new game is presented and so on till a termination case is reached. Such scenario are endless, and can be found in various subjects. Thus the application of the research is wide. The papers [28], [20] which show applications of similar research to different fields

8

Conclusion

There are main two perspective from where the research can be viewed at

1. For a zero sum multi-stage game , coEvolutionary algorithm can be used to find a solution. However this is with a caution that the objective function should have a common set of optimal solution. For a disjoint stage matrices where there is no presence of common solution set, Agent may end up performing worst in the game.

2. Coevolutionary algorithm can be applied to the Game theory and a solution can be found. During the research an analysis of the algorithm led to the development of an equation of computational time with respect to mutation rate, crossover rate, Interaction, and population. This equation is helpful in cases of applications which takes high computation. A range can be narrowed down using the equation which is helpful for application of the GA or CoEA that may take weeks for computation otherwise.

9

Summary

Game of fight is analysed in the research. A constrained world where two agents interact with each other just with horizontal mobility. The translation of the game led to a development of zero sum multistage game. Analysis through game theory shows, The individual stage solution to the developed game is an infinite solution set. This is then reiterated through Application of coevolutionary algorithm to the individual games. However the main motive of the research is to present a solution of multistage game. This is done by using a composite function as an optimisation function. This optimised solution is compared with the individual stages where solution set was infinite. However in case of Multistage game evolution, Game evolved to a probability set of [0.5,0.5,0,0]. Further, An analysis is done on the algorithm to identify what will be a best set of parameters which will converge the algorithm with possibly least computation time. An novel equation has been developed for computation time. A range of parameters is identified. Later, a new game is developed and new set of parameters are applied to it. Convergence occurs with the parameters, this means the parameters can be reused for new game setting as well.

Bibliography

[1] Bendor, J., Mookherjee, D., & Ray, D. (2001). Reinforcement learning in repeated interaction games. *Advances in Theoretical Economics*, *1*(1).

[2] Chess, D. M. (1988). Simulating the evolution of behavior: the iterated prisoners dilemma problem. *Complex systems*, *2*(6), 663–670.

[3] Cressman, R. (1995). Evolutionary stability for two-stage hawk-dove games. *The Rocky Mountain Journal of Mathematics*, 145–155.

[4] Dawid, H. & Mehlmann, A. (1996). Genetic learning in strategic form games. *Complexity*, *1*(5), 51–59.

[5] Farooqui, A. D. & Niazi, M. A. (2016). Game theory models for communication between agents: a review. *Complex Adaptive Systems Modeling*, *4*(1), 13.

[6] Freund, Y. & Schapire, R. E. (1996). Game theory, on-line prediction and boosting. In *COLT*, volume 96, (pp. 325–332). Citeseer.

[7] Fudenberg, D. & Tirole, J. (1991). Game theory, 1991. *Cambridge, Massachusetts*, *393*(12), 80.

[8] Golbeck, J. (2002). Evolving strategies for the prisoners dilemma. *Advances in Intelligent Systems, Fuzzy Systems, and Evolutionary Computation*, *2002*, 299.

[9] Hassan, A. S. & Rafie, M. M. (2010). A survey of game theory using evolutionary algorithms. In *2010 International Symposium on Information Technology*, volume 3, (pp. 1319–1325). IEEE.

[10] Haywood Jr, O. (1954). Military decision and game theory. *Journal of the Operations Research Society of America*, *2*(4), 365–385.

[11] Holler, M. J. (2002). Classical, modern, and new game theory/klassische, moderne und neue spieltheorie. *Jahrbücher für Nationalökonomie und Statistik*, *222*(5), 556–583.

[12] Ismail, I. A., El Ramly, N. A., El Kafrawy, M., & Nasef, M. M. (2007). Game theory using genetic algorithms. In *World Congress on Engineering*, (pp. 61–64).

[13] Littman, M. L. (1994). Markov games as a framework for multi-agent reinforcement learning. In *Machine learning proceedings 1994* (pp. 157–163). Elsevier.

[14] Nagrale, S. (2018). Game theory. `http://gametheorysumedh.blogspot.com/`.

[15] Nash, J. (1951). Non-cooperative games. *Annals of mathematics*, 286–295.

[16] Ni, Z. & Paul, S. (2019). A multistage game in smart grid security: A reinforcement learning solution. *IEEE transactions on neural networks and learning systems*.

[17] O'Dwyer, D. W. & OBrien, E. J. (1970). Genetic algorithm encoding probabilities & population size. *WIT Transactions on Information and Communication Technologies*, *20*.

[18] Pace, M. (2009). How a genetic algorithm learns to play traveler's dilemma by choosing dominated strategies to achieve greater payoffs. In *2009 IEEE Symposium on Computational Intelligence and Games*, (pp. 194–200). IEEE.

[19] Pearce, D. G. (1984). Rationalizable strategic behavior and the problem of perfection. *Econometrica: Journal of the Econometric Society*, 1029–1050.

[20] Peng, X., Lei, X., & Liu, K. (2014). Compensate information from multimodal dynamic landscapes: An anti-pathology cooperative coevolutionary algorithm. In *2014 IEEE Congress on Evolutionary Computation (CEC)*, (pp. 2578–2584). IEEE.

[21] Petrosyan, L., Chistyakov, S., & Pankratova, Y. (2017). Existence of strong nash equilibrium in repeated and multistage games. In *2017 Constructive Nonsmooth Analysis and Related Topics (dedicated to the memory of VF Demyanov)(CNSA)*, (pp. 1–3). IEEE.

[22] Popovici, E., Bucci, A., Wiegand, R. P., & De Jong, E. D. (2012). Coevolutionary principles. *Handbook of natural computing*, 987–1033.

[23] Rosin, C. D. & Belew, R. K. (1997). New methods for competitive coevolution. *Evolutionary computation*, *5*(1), 1–29.

[24] Sim, K.-B., Lee, D.-W., & Kim, J.-Y. (2004). Game theory based coevolutionary algorithm: a new computational coevolutionary approach. *International Journal of Control, Automation, and Systems*, *2*(4), 463–474.

[25] Sudhoff, S. Purdue: ee630 a manual example using canonical gas. `https://engineering.purdue.edu/~sudhoff/ee630/Lecture02.pdf`.

[26] Tadelis, S. (2005). Stanford, Lecture Notes: Econ 160, Week 5. URL: `http://faculty.haas.berkeley.edu/stadelis/Game%20Theory/econ160_week5.pdf`.

[27] Von Neumann, J. & Morgenstern, O. (2007). *Theory of games and economic behavior (commemorative edition)*. Princeton university press.

[28] Zhu, Q. & Rass, S. (2018). On multi-phase and multi-stage game-theoretic modeling of advanced persistent threats. *IEEE Access*, *6*, 13958–13971.